Attorney Docket No.: YO999-011 (8728-255)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Challenger et al. Examiner: William D. Hutton

SERIAL NO.: 09/283,561 Group Art Unit: 2178

FILED: April 1, 1999 Dated: August 2, 2005

FOR: METHOD AND SYSTEM FOR EFFICIENTLY CONSTRUCTING AND CONSISTENTLY PUBLISHING WEB DOCUMENTS

Mail Stop: Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Enclosed please find APPELLANTS' APPEAL BRIEF.

The amount of $500.00 is hereby authorized to be charged Deposit Account No. 50-0510/IBM (Yorktown Heights). Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and/or 1.17, at any time during the pendency of this application, or credit any overpayment of such fee(s), to Deposit Account No. 50-0510/IBM (Yorktown Heights).

Respectfully submitted,

Frank V. DeRosa
Reg. No. 43,584
Attorney for Applicant(s)

F. CHAU & ASSOCIATES, LLC
130 Woodbury Road
Woodbury, NY 11797
Tel: (516) 692-8888
Fax: (516) 692-8889

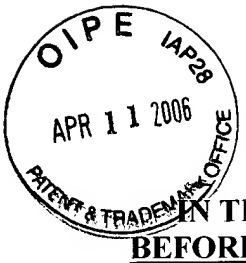
C

ERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

I hereby certify that this correspondence and accompanying documents are being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Mail Stop Appeal Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

Dated: 8/2/05

Frank V. DeRosa



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants: Challenger et al.

Examiner: William D. Hutton

Serial No: 09/283,561

Group Art Unit: 2178

Filed: April 1, 1999

Docket: Y0R91999011US1 (8728-255)

For: **METHOD AND SYSTEM FOR EFFICIENTLY CONSTRUCTING AND CONSISTENTLY PUBLISHING WEB DOCUMENTS**

APPEAL BRIEF

Appeal from Group 2178
F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellants

TABLE OF CONTENTS **Page(s)**

I. INTRODUCTION

This Appeal is from a Final Office Action mailed on January 27, 2005 (hereinafter, referred to as the "Final Action") finally rejecting claims 16-26, 42-53, 55-60, and 75-81 of the above-identified application. The Appellants commenced this Appeal by a Notice of Appeal filed on May 2, 2005, and hereby submit this Appeal Brief in furtherance of the Appeal.

II. REAL PARTY IN INTEREST

The real party in interest for the above-identified application is International Business Machines Corporation, the assignee of the entire right, title and interest in and to the subject application by virtue of an assignment of recorded in the U.S. Patent and Trademark Office.

III. RELATED APPEALS AND INTERFERENCES

There are no Appeals or Interferences known to Applicant, Applicant's representatives or the Assignee, which would directly affect or be indirectly affected by or have a bearing on the Board's decision in the pending Appeal.

IV. STATUS OF CLAIMS

Claims 16-26, 42-53, 55-60, and 75-81 are pending, stand rejected and are under appeal. The claims are set forth in the attached Appendix. Claims 16, 42, 53, and 75 are independent claims, claims 17-26 depend from claim 16, claims 43-52 depend from claim 43, claims 55-60 depend from claim 53, and claims 76-81 depend from claim 75.

V. STATUS OF AMENDMENTS

No claim amendments have been filed or entered subsequent to the Final Action.

No claim amendments have been filed or entered subsequent to the Final Action.

VI. SUMMARY OF CLAIMED SUBJECT MATTER

In general, the claimed inventions are directed to methods for computerized *publication* of publishable objects (e.g., documents such as web pages). The process of "*publishing*" an object includes making the object visible to the public or a community of users. For example, in generating Web content, publishable Web pages (known as "servables") may be published on Web sites. For illustrative purposes, the claimed subject matter will be described with reference to exemplary embodiments described in Appellants specification (hereinafter, Spec.) and accompanying figures, although nothing herein shall be construed as unduly limiting the scope of the claimed subject matter.

The claimed inventions provide methods for *publishing* objects (e.g., web pages) based on one or more *consistency constraints* that enable publishable objects to be published in a consistent manner and in particular, consistent publication of publishable objects that are constructed with *fragments*. As discussed in Appellants' specification, a *fragment* is an *object* which is used to construct a *compound object*. An *object* is an entity which can either be published or is used to create something which is publishable. *Objects* include both *fragments* and *compound objects*. A *compound object* is an *object* constructed from one or more *fragments*. A publishable *object* may include a *fragment* which in turn includes another *fragment*, etc. (See Spec., page 10, lines 1-15).

In one exemplary embodiment of the invention, when Web pages are constructed using *fragments*, the Web pages should be published in a consistent manner as Web pages are created

or updated so that the Web site will look consistent. The concept of "*publishing consistency*" is generally explained with reference to the illustrative example of FIG. 1 and corresponding description in Spec. FIG. 1 depicts three servables (publishable objects) **S1**, **S2** and **S3**, where **S1** includes *fragments **f1*** and ***f2***, **S2** includes *fragment* **f1** and **S3** includes *fragment* **f2**. In the example, servables **S1** and **S2** include a *common fragment* **f1**. If *fragment* **f1** is updated/changed, updated versions of **S1** and **S2** should be published concurrently, otherwise, the Web Site will look inconsistent. Further, since servables **S1** and **S3** include a common *fragment* **f2**, if *fragment* **f2** is *updated/changed*, updated versions of both **S1** and **S3** should be *published* concurrently to maintain consistency of the web site. By way of further example, if both *fragments* **f1** and **f2** are *updated/changed*, updated versions of **S1, S2**, and **S3** could be *published* concurrently to maintain *consistency*, even though **S2** and **S3** do not include a *common fragment*. (See Spec., page 11, lines 1-20).

In accordance with the invention, the process of "*publishing*" an *object* is decoupled from the processes of *creating* or *updating* the *object* and the process of *publishing* generally occurs after the *object* has been *created* or *updated*. (See, e.g., Spec., page 10, lines 11-17). For example, FIG. 2 of the Spec. is a flow diagram that illustrates a method for efficiently constructing objects and publishing objects. In FIG. 2, the process steps (100, 110, 120 and 130) include methods for determining a set of objects ***S*** that are affected by a change to one or more objects (set of objects ***C***) in a set of object, and then determining an *order in which the objects in **S** can be updated in an efficient manner*. (See Spec., page 12, line 20 through page 14, line 16).

Moreover, in FIG. 2, the process step (140) includes methods for *publishing* the objects in ***S*** after such objects are *updated*. The *publishing* process can be performed such that all objects

in ***S*** are published concurrently after the objects are *updated.* This "all-at-once" publishing process avoids consistency problems because all updated objects are published after they are updated. (Spec., page 14, lines 17-18).

In another embodiment, an "*incremental publication*" process can be implemented, which reduces the number of objects that need to be published in a single atomic action. For incremental publishing, methods are implemented to *determine an order in which updated objects can be published according to one or more specified consistency constraints such that incremental publishing maintains consistency.* With incremental publishing, one updated object or one group of updated objects in ***S*** can be published before another updated object or another group of updated objects in ***S***. (See Spec. page 14, line 15 thorugh page 15, line 16). To support incremental publication, methods are provided for determining groups of objects that can be published together. (See, generally, Fig. 5 and corresponding description in Spec.).

Independent **Claim 16** broadly embodies features as described above and recites:

A method for publishing a plurality of objects comprising the steps of:

providing a plurality of objects, including compound objects;

partitioning at least some of the plurality of objects into a plurality of groups such that if two compound objects are constructed from at least one common changed fragment, then the compound objects are placed in a same group; and

publishing all objects belonging to a same group together.

In general, **Claim 16** recites a method for publishing objects in accordance with a consistency constraint whereby a group of objects having a *common fragment* whose value has changed are published together. The method includes *providing a plurality of objects, including*

compound objects. As noted above, an *object* is an entity which can either be published or is used to create something which is publishable, wherein o*bjects* include both *fragments* and *compound objects,* wherein a *compound object* is an *object* constructed from one or more *fragments*. (See Spec. p. 10, lines 1-15).

The method further includes *partitioning at least some of the plurality of objects into a plurality of groups.* The *partitioning* is performed based on a consistency constraint whereby *two compound objects will be placed in a same group if the two objects are constructed from at least one common changed fragment.* (See, e.g., Spec., p. 11, lines 1-20) Thereafter, all *objects belonging to a same group are published together.*

Furthermore, in another exemplary embodiment as recited in **Claim 17**, the publishing step is performed such that *for at least two of the plurality of groups, publishing all objects belonging to a first group before publishing any objects belonging to a second group.* (See, Spec. P. 14, line 15 ~ p. 15, line 10).

In another exemplary embodiment as recited in **Claim 18,** the step of publishing may include *delaying publication of a first object until a second object which is referenced by the first object is published.* (See, Spec. P. 14, line 15 ~ p. 15, line 16).

The subject matter of Claims 42, 43 and 44 is similar to that of claims 16, 17 and 18, respectively, so the above summary description of such claims is applicable.

Independent **Claim 53** broadly embodies features as described above and recites:

A method for publishing a plurality of objects comprising the steps of:

providing a plurality of objects;

constructing at least one graph, the at least one graph including nodes representing at least some of the plurality of objects and edges for connecting nodes having relationships, at least some of the edges being derived from at least one consistency constraint;

finding at least one strongly connected component in the at least one graph; and

publishing a set of objects belonging to a same strongly connected component group.

The method of Claim 53 can be explained with reference to exemplary embodiment of FIG. 5 and other relevant description in the Spec. In general, the invention of claim 53 provides methods that implement graph data processing techniques for determining groups of objects that can be published together based on one or more specified consistency constraints. More specifically, the method includes *providing a plurality of objects* (as noted above, an *object* can be a *fragment* or a *compound object* (see Spec. p. 10, lines 1-15).).

The method of claim 53 further includes *constructing a graph that includes nodes which represent at least some of the plurality of objects. The graph includes edges for connecting nodes having relationship. At least some of the edges are derived from at least one consistency constraint.* In the exemplary embodiment of FIG. 5, step 410 depicts a method for creating a consistency graph in which the graph vertices/nodes represent servables (publishable objects) and edges between vertices/modes. The edges between the vertices/nodes comprise *"consistency edges"* between vertices/nodes (objects). The *consistency edges* are based on one or more *consistency constraints* that imply an order in which the objects should be published to achieve publishing constancy. In other words, the consistency edges do not imply an order in which objects are created/updated, only an order in which the objects are published (see, e.g., Spec. p. 17, lines 4-12).

The method of **Claim 53** further includes *finding at least one strongly connected component in the at least one graph,* and then *publishing a set of objects belonging to a same strongly connected component group.* In the exemplary embodiment of FIG. 5, steps 420 and 430 depict methods for *finding strongly connected components* using graph traversal methods, wherein each strongly connected component corresponds to a set of servables (objects) that can be published together and publishing the objects according to strongly connected components (see, e.g., Spec. p. 17, line 13 – p. 18, line 10).

In exemplary embodiments of the invention as recited in **Claims 55 and 56**, the claimed "finding" step of **Claim 53** comprises *topologically sorting at least part of the graph*, wherein the claimed publishing step of **Claim 53** *comprises examining objects in an order defined by the topological sort,* and *when an unpublished object is examined, publishing the unpublished object together with all objects belonging to a same strongly connected component.*(See, e.g., **Claim 56)**

In another exemplary embodiment of the invention as recited in **Claim 57**, *the at least one consistency constraint* (in **Claim 53**) *includes delaying publication of a first object before a second object which is referenced by the first object is published.* For example, as recited in **Claim 58**, delayed publication is warranted *wherein the first and second objects include Web pages and at least one edge between the objects corresponds to at least one hypertext link.*

In another exemplary embodiment of the invention as recited in **Claims 59**, *an edge exists from a first object to a second object in at least one of the at least one graphs if the second object has a reference to the first object*. More specifically, in the invention of **Claim 53**, a **consistency edge** may exist between two servables if there is a hypertext link from one servable to the other.

In another exemplary embodiment of the invention as recited in **Claim 60**, *the at least one constancy constraint* (in **Claim 53**) *includes publishing two compound objects together if the two compound objects are both constructed from at least one common changed fragment.* By way of specific example, *consistency edges* can be used to indicate that two servables both embed a common fragment whose value has changed and thus are to be published concurrently.

The subject matter of Claims 53, 55, 56, 57, 58, 59, and 60 is similar to that of claims 75, 76, 77, 78, 79, 80 and 81, 17 and 18, respectively, so the above summary description of such claims is applicable.

VII. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Anticipation Rejections

A. Claims 16, 17, 42 and 43 stand rejected under 35 U.S.C. 102(e) as being anticipated by the publication by Darnell, et al., "Using Macromedia Dreamweaver 1.2", Chapter 8, pp. 117-123, 1998

Obviousness Rejections

B. Claims 18, 19, 44 and 45 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Darnell.

C. Claims 20-22 and 46-48 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Darnell in view of U.S. patent No. 6,199,082 to Ferrel.

D. Claims 23-26, 49-53 and 55-60 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Darnell, in view of Ferrel and further in view of Cormen, et al., "Introduction to Algorithms, pp 477-493.

VIII. ARGUMENTS

ANTICIPATION REJECTIONS

For a claim to be anticipated under 35 U.S.C. § 102, it is well settled that anticipation is established only when a single prior art reference discloses, expressly or under principles of inherency, each and every element of the claimed invention. RCA Corp. v. Applied Digital Data Syst., Inc., 730 F.2d 1440, 1444, 221 USPQ 385, 388 (Fed. Cir. 1984); See, also Scripps Clinic & Research Found. v. Genentech Inc., 927 F.2d 1565, 1576, 18 U.S.P.Q.2d. 1001, 1010 (Fed. Cir. 1991). The identical invention must be shown in as complete detail as is contained in the claim. (See MPEP § 2131). The single prior art reference must disclose all of the elements of the claimed invention functioning essentially in the same manner. See, e.g., Shanklin Corp. v. Springfield Photo Mount Corp, 521 F.2d 609 (1st Cir. 1975).

A. Darnell Does Not Anticipate Claims 16, 17, 42 or 43

Appellants will show that Examiner's reliance on Darnell to support the anticipation rejections is misplaced, and that Darnell is legally deficient to establish *prima facie* case of anticipation against any of claims 16, 17, 42 and 43. The grounds for the anticipation rejection are set forth on pages 4-7 of the Final Action. The Examiner provides a detailed analysis to support the anticipation rejections of claim 16 and 17, and then relies on the same rationale for rejection claims 42 and 43 (see Final Action, page 7). Accordingly, for purposes of this Appeal, Appellants will address the rejections of claims 16 and 42 and for claims 17 and 43 together.

In the Final Action, in formulating the anticipation rejections, the Examiner acknowledges that Darnell does not expressly disclose various novel elements of the inventions

of claims 16, 17, 42 and 43, but the Examiner relies instead on the theory of "inherency", contending that Darnell inherently discloses such novel elements. Although anticipation may be based upon inherency, in order to rely on the theory of inherency, the Examiner *must* provide some *basis-in-fact* or technical reasoning which reasonably supports the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art. See Ex parte Levy, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). In the case at bar, Appellants will demonstrate that the Examiner has not, and cannot, meet his burden of establishing anticipation in view of Darnell based on a theory of inherency. The following discussion will proceed with a brief description of Darnell, followed by an explanation as the impropriety of the anticipation rejections.

Summary of Darnell

In general, Darnell discloses a web page development and publishing tool (referred to as "Dreamweaver") which includes methods for (i) creating and editing a library of reusable page segments, (ii) inserting library items on a Web page; and (iii) updating all library items on all web site pages (see, Darnell, page 117). More specifically, Darnell discloses a method for creating "web page parts" that can be stored as library items and managed using a library palette function (see Darnell, page 118). The library palette includes a function that allows a user to select a library item to be added to a given web page (see Darnell, page 121), and an edit function that allows a user to edit a library item (see Darnell, page 120).

Darnell discloses a "*many pages, one part, one update*" function that allows a user to **update** the pages of an entire site with an edited library item. In particular, Darnell discloses an **update process** in which the *Dreamweaver* tool will search through the pages of the web site for

library items, and **update** any it finds with the current contents from the library (see Darnell, page 120). Darnell does not disclose specific details regarding the manner in which the web pages are searched and updated, much less the manner in which updated web pages are published.

(1) Darnell Does Not Anticipate Claims 16 and 42

Appellants contend that Darnell is legally deficient to establish a *prima facie* case of anticipation against claims 16 and 42. In particular, it is contended that Darnell does not disclose or suggest, either expressly or inherently, a *method for publishing a plurality of objects* which includes (i) ***partitioning*** *at least some of the plurality of objects into a plurality of groups such that if two compound objects are constructed from at least one common changed fragment, then the compound objects are placed in a same group*, much less (ii) ***publishing*** *all objects belonging to a same group together*, as essentially recited in claims 16 and 42.

The anticipation rejection for claims 16 and 42 is set forth on pages 4-6 of the Final Action. In formulating the rejections, the Examiner equates the claimed "fragments" with Darnell's "web page parts" contending that the claimed "changed fragments" are Darnell's "edited library items". The Examiner acknowledges that Darnell does not expressly disclose the claimed "partitioning" step, but relies instead on the theory of "inherency", contending that Darnell inherently discloses the "partitioning" step. In particular, for ease of reference, the Examiner's rational for inherency as set forth on pages 5-6 of the Final Action, is as follows:

> ... The edited "library items" are the "changed fragments." Dreamweaver allows the user to edit multiple "library items" and update all web pages at once, as clearly indicated in the cited text. In this "atomic" update of web pages, Dreamweaver searches through all of the web pages for the website and updates all of those pages containing the edited "library items", as clearly indicated in the text. In order to do this, Dreamweaver must ***inherently*** "place compound objects having a common changed fragment into a same group" in that, for each edited "library item,"

> Dreamweaver will determine which web pages of the website contain said "library item." In doing this, Dreamweaver has "[partitioned] at least some of the plurality of objects into a plurality of groups such that if two compound objects are constructed from at least one common changed fragment, then the compound objects are placed in a same group."

Furthermore, with regard to the claimed "publishing" step of claims 16 and 42, the Examiner continues on page 6 of the Final Action as follows:

> ... Darnell discloses [on pages 117-123] that Dreamweaver allows users to collaboratively work on the content of web pages for a website and allows said users to publish their website, as clear indicated in the cited text. In doing this, Dreamweaver sets up a "working site" and a "final site". After all final edits to the "library items" and each individual web page have been made, the website is sent to the web server. Thus, Darnell discloses "publishing all objects belonging to a same group together."

However, for reasons explained hereafter, Appellants respectfully contend that the above analyses are *fundamentally flawed*, both technically and legally, on various levels, and do not support the anticipation rejections for the following reasons.

(a) *The Anticipation Analysis is based on Improper Claim Interpretation*

For example, Appellants contend that Examiner's anticipation analysis is based, in part, on an *unreasonable parsing and interpretation* of the claim language. For example, the Examiner offers an erroneous interpretation of the claimed "partitioning" step based on the object updating methods of Darnell. Essentially, the Examiner When properly construed, however, for reasons explained hereafter, the claimed "partitioning" step relates to publishing objects for purposes of achieving consistency in the publication, not updating objects. partitioning" step is within the context of publishing objects for purposes of achieving consistency in the publication,

In particular, as can be readily gleaned from the above cited portions of the anticipation analysis regarding "partitioning", the Examiner points to Darnell's teaching of an "atomic" update of web pages (i.e., update all web pages at once), wherein Dreamweaver searches through all of the web pages for the website and *updates* all of the web pages containing the edited "library items." In concluding that Darnell discloses the claimed "partitioning" step, the Examiner essentially avers that **"in order to do this [i.e., search and update the web pages], Dreamweaver must *inherently* 'place compound objects having a common changed fragment into a same group' . . ."** Essentially, the Examiner construes the claimed "partitioning" step as a process of grouping/dividing a website into groups that include web pages having the same edited library items" and make the required changes to the web pages (see also, Final Action, p. 6)

In other words, it is apparent that the Examiner has construed the claimed "partitioning" step for grouping objects in the context of grouping objects for purposes of searching and updating web pages. This is further supported by evident from Examiner's analysis (as noted above) wherein Examiner contends that "After all final edits to the "library items" and each individual web page have been made, the website is sent to the web server" for publishing. In other words, Examiner's analysis distinguishes from the atomic update process wherein web pages are searched and updated with edited library items, and the publishing which occurs after all web page changes have been made." Thus, Darnell discloses "publishing all objects belonging to a same group together."

On a fundamental level, the "updating" methods of Darnell are distinct from, and unrelated to, the claimed "partitioning" step. As noted above, in accordance with the invention,

the process of "*publishing*" an *object* is decoupled from the processes of *creating* or "*updating*" the *object* and the process of *publishing* generally occurs after the *object* has been *created* or *updated*. In the invention of claim 16 and 42, ***partitioning*** *objects into a plurality of groups such that if two compound objects are constructed from at least one common changed fragment, then the compound objects are placed in a same group.* In other words, with the partitioning, compound objects having a common changed fragment are placed in the same group. When properly construed, the partitioning is applied to updated objects compound objects with a common changed fragment are placed in the same group. Indeed, as explained Section VII above, the step of *partitioning of objects having common changed fragments into a plurality of groups* allows for achieving consistency in the publication, wherein ***objects belonging to a same group are published together***.

In this regard, the Examiner's reliance on Dreamweaver's "update" process as inherently disclosing the claimed "partitioning" is flawed on a fundamental level in that the claimed "partitioning" step is within the context of publishing objects for purposes of achieving consistency in the publication, not updating objects. In this regard, even assuming *arguendo* that Examiner's analysis is correct, Darnell discloses *at most* partitioning objects into groups for purposes of updating the objects, not for publishing objects in the same group together.

In fact, Darnell is essentially devoid of any specific disclosure related to "publishing", much less publishing groups of objects that are determined based on a partitioning step. In this regard, the Examiner goes to great pains to explain (page 6 of the Final Action) how Darnell discloses the claimed "publishing all objects belonging a same group together". In particular,

with regard to the claimed "publishing" step of claims 16 and 42, the Examiner continues on page 6 of the Final Action as follows:

> ... Darnel discloses [on pages 117-123] that Dreamweaver allows users to collaboratively work on the content of web pages for a website and allows said users to publish their website, as clear indicated in the cited text. In doing this, Dreamweaver sets up a "working site" and a "final site". After all final edits to the "library items" and each individual web page have been made, the website is sent to the web server. Thus, Darnell discloses "publishing all objects belonging to a same group together."

However, Examiner's analysis is merely conclusory and based on a self serving strained interpretation of Darnell that is not supported by specific citation to the teachings of Darnell. Indeed, the crux of Examiner's argument outlined above is essentially that *all updated objects belong to a same group of objects* and that *after all objects are updated, they are published as a group*. However, this argument is simplistic, misses the point and mischaracterizes the claimed publishing step. To begin, the Examiner's analysis fails to consider the proper context of the claim in that the partitioning of objects into a plurality is groups is such that *objects having common changed fragments* are grouped together. In other words, in the proper context of the claimed invention, a group of objects includes objects that have common changed fragments. In contrast, based on Examiner's interpretation, a group of objects includes all objects that have been updated. In this regard, the anticipation rejection, the Examiner simply ignores of group of object is object groups for purposes of achieving consistency in the publication, wherein ***objects belonging to a same group are published together.***

(b) ***The anticipation rejections are not supported under principles of inherency***

For instance, as will be explained in detail below, the Examiner has offered no evidence or basis-in-fact to support the purported "inherent" teachings of Darnell. Moreover, other than

pure conjecture and speculation, the Examiner has offered no legally sufficient rational or technical reasoning to support the anticipation rejection based on inherency.

As noted above, to rely on the doctrine of inherency, the Examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art. In order for something to be "inherent" in a disclosure it must be the necessary and only reasonable construction to be given to the disclosure, that is, the result claimed must inevitably occur. The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. In re Rijckaert, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) Inherency is not established by possibility or probability; for a result to be deemed inherent, it must invariably exist or occur.

In the case at bar, the Examiner has offered no evidence or basis-in-fact to support the purported "inherent" teachings of Darnell with regard to "partitioning". In fact, the Examiner initially contended that Darnell did in-fact inherently disclose the claimed "partitioning" step because the Examiner had "confirmed this in a discussion of this case with a fellow examiner who has worked extensively with Dreamweaver 1.2" (see, page 26 of the Final Action). In response to this assertion, Applicants filed a Response to the Final Action filed, requesting that Examiner Hutton provide a sworn affidavit attesting to his specific knowledge as to the functionalities of Dreamweaver 1.2, and explaining in detail how the claimed "partitioning" is included in Dreamweaver 1.2 or inherently disclosed in Darnell. However, in the Advisory Action of 5/20/05, Examiner Hutton admits that he "does not have any 'specific knowledge' as to the functionalities of Dreamweaver 1.2" and that the rejections are not based on facts within

his personal knowledge. Instead, Examiner Hutton acknowledges that he "was simply using a common sense approach ."

However, the Examiner cannot establish a prima facie case of anticipation based on inherency using a "common sense approach." In order to for something to be "inherent" in a disclosure, it must be the necessary and only reasonable construction to be given to the disclosure, that is, the result claimed must inevitably occur. The Examiner essentially contends that because Dreamweaver searches through all of the web pages for the website an updates all of those pages containing the edited library items, that Dreamwever must inherently perform the claimed partitioning step to place compound objects having a common changes fragment into a same "group." However, it is glaringly apparent that the Examiner's detailed analysis as to the "inherent" functionalities of Dreamweaver is based simply on surmise and conjecture in hindsight view of teachings of Applicants specification. Indeed, Darnell merely discloses, without more, that the update process involves "searching through the pages on the web site for library items and updating any library items that are found using the current contents from the library (see page 122). This process can be implemented many ways. For instance, the update process can sequentially select each page (in any order), determine if the page has library items to be updated, and then update the library items for that page as needed, and repeat this process for all pages. In such instance, there is no need to group the objects having common changed fragments. Examiner has not explained why this is the only manner in which the update process can be performed. If this is true, the Examiner should be able to find a myriad of prior art references that describe this "one and only method."

For at least the above reasons, it is glaringly apparent that Examiner has not met his burden of establishing anticipation against claims 16 and 42 based on Darnell.

(2) Darnell Does Not Anticipate Claims 17 and 43

In view of the above, it is clear that Darnell does not disclose a publishing step that includes, *for at least two of the plurality of groups, publishing all objects belonging to a first group before publishing any objects belonging to a second group*, as recited in claims 17 and 43. Indeed, the rejection of claims 17 and 43 is based on the Examiner's finding of the claimed "partitioning" step based on "inherency". In this regard, the anticipation rejections of claims 17 and 43 are legally deficient on their face.

OBVIOUSNESS REJECTIONS

In rejecting claims under 35 U.S.C. 103, the Examiner bears the initial burden of presenting a prima facie case of obviousness. In re Rijckaert, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a prima facie case of obviousness is only satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988). The test for obviousness is what the combined teachings of the applied prior art references would have suggested to one of ordinary skill in the art. In re Keller, 642 F.2d 413, 435; 208 U.S.P.Q. 871, 881 (CCPA 1981). The suggestion to combine the references should come from the prior art, and the Examiner cannot use hindsight gleaned from the invention itself to pick and choose among related prior art references to arrive at the claimed invention. In re Fine, 837 F.2d at 1075. If the Examiner fails to establish a prima facie case, the rejection is improper and must be overturned. In re Rijckaert, 9 F.3d at 1532 (citing In re Fine,

837 F.2d at 1074).

Appellants will demonstrate that the obviousness rejections B, C and D as listed in Section VII above are legally deficient and unsupportable by the teachings of the cited combinations of references. Moreover, as will be readily apparent to the Board, the obviousness rejections are premised not only on improper hindsight reasoning in view of Appellants specification, but also "inherency" arguments which, as discussed above, are based purely on surmise and conjecture. In this regard, it is contended that the Examiner has not met his burden of establishing a *prima facie* case of obviousness to support the rejections B C and D.

B. Darnell is Legally Deficient to Establish a Prima Facie Case of Obviousness Against Claims 18, 19, 44 and 45

Claims 18, 19, 44 and 45 stand rejected as being unpatentable over Darnell for the reasons set forth in the Final Action. Rather than specifically address this rejection, it is suffice to say that the obviousness rejection is invalid at least for the same reasons given above for Claims 16 and 42. Indeed, because Claims 18 and 19 incorporate the elements of Claim 16 by virtue of dependency, and claims 44 and 45 incorporate the elements of Claim 42 by virtue of dependency and since the rejection of Claims 16 and 42 are based on an improper finding of anticipation based on Darnell, the Final Action fails at the very least to demonstrate how the teachings of Darnell meets or renders obvious elements of these claims.

C. The Combination of Darnell and Ferrel is Legally Deficient to Establish a *Prima Facie* Case of Obviousness Against Claims 20-22 and 46-48

Rather than specifically address this rejection, it is suffice to say that the obviousness rejection is invalid at least for the same reasons given above for Claims 16 and 42. Indeed,

because Claims 20-22 incorporate the elements of Claim 16 by virtue of dependency, and claims 46-48 incorporate the elements of Claim 42 by virtue of dependency and since the rejection of Claims 16 and 42 are based on an improper finding of anticipation based on Darnell, the Final Action fails at the very least to demonstrate how the combined teachings of Darnell and Ferrel meets or renders obvious elements of these claims.

D. The Combination of Darnell, Ferrel and Cormen is Legally Deficient to Establish a *Prima Facie* Case of Obviousness Against Claims 23-26, 49-53 and 55-60 and 75-81

In the following discussion, Appellants will specifically address the obviousness rejections with regard to independent claims 53 and 71. With regard to claims 23-26 and 49-52, it is suffice to say that the obviousness rejections are invalid at least for the same reasons given above for claims 16 and 20, from which claims 23-26 depend, as well as claims 42 and 48, from which claims 49-52 depend.

(1) Claims 53 and 71 are not obvious in view of Darnell, Ferrel and Cormen

The obviousness analysis to support the rejection of claim 53 is set forth on pages 17-20 of the Final Action. The Examiner relies on the same rational for rejecting claim 71, so these claims will be addressed together.

Appellant contend that the combination of Darnell, Ferrel and Cormen does not teach or fairly suggest the inventions of claims 53 or 71 as a whole. For instance, claim 53 recites a method for publishing a plurality of objects comprising the steps of:

providing a plurality of objects;

constructing at least one graph, the at least one graph including nodes representing at least some of the plurality of objects and edges for connecting nodes having relationships, at least some of the edges being derived from at least one consistency constraint;

finding at least one strongly connected component in the at least one graph; and

publishing a set of objects belonging to a same strongly connected component group.

Overall, the combination of Darnell, Ferrel and Cormen does not fairly suggest the invention of claim 53. As discussed above, claim 53 is directed to a method for publishing objects, wherein objects are grouped together based o n consistency constraints such that grouped objects can be published together. In the invention of claim 53, the objects are grouped using a graph process in which consistency constraint are included as edges between object (nodes) in the graph and using graph processing find strongly connected components (which is a subset of vertices of any digraph and any edges between them that forma strongly connected graph where for each vertex there exists a path from the vertex to every other vertex..

In support of the rejection, the Examiner relies on Darnell as generally teaching a method for publishing a plurality of objects. In this regard, Examiner's reliance on Darnell is somewhat irrelevant given that Darnell does not disclose the claimed step of "constructing" a graph. (see Final Action, p. 17) . The Examiner relies on Ferrel as teaching *constructing at least one graph, the at least one graph including nodes representing at least some of the plurality of objects and edges for connecting nodes having relationships, at least some of the edges being derived from at least one consistency constraint* The Examiner's rationale for this is set forth on page 18 of the Final Action. In particular, the Examiner cites two lines (Col. 9, lines 30-31) of Ferrel, which states that the "natural way of storing related and ordered objects is in a data structure, such as an acyclic graph". The Examiner contends that acyclic graphs inherently include "nodes"

representing objects and "edges" for connecting nodes having relationships. The Examiner interprets the term "consistency constraint: as being "*any relationship or order that is imposed upon objects of web pages that will affect the publication of said web pages*". The Examiner then concludes that "*edges of an acyclic graph inherently are derived from a consistency constraint* ..." The Examiner relies on this analysis to conclude that "it would have been obvious ... to modify the method of Darnell with the teachings of Ferrel to derive the claimed constructing step. This analysis is fundamentally flawed for various reasons.

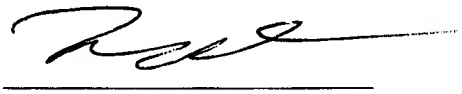
For instance, this analysis is based purely on hindsight reasoning in view of Appellants disclosure as there is simply no suggestion in either Darnell or Ferrel of a method for publishing objects in which a graph is constructed with *at least some of the edges being derived from at least one consistency constraint* As noted above, Darnell does not disclose a method for publishing, much less a method for publishing objects based on consistency constraints. At most Darnell discloses a method of updating Web pages with edited library items. Moreover, even assuming arguendo that graphs comprising nodes and edges are well known, other than through hindsight reasoning, there is not basis for Examiner's conclusion that Ferrel teaches a graph having nodes and edges, wherein *at least some of the edges being derived from at least one consistency constraint*. Although acyclic graphs may be known to have ordered (directed) edges, this is not the same as edges derived from *consistency constraints*, as claimed.

Therefore, other than the simplistic, strained interpretation offered by the Examiner, there is simply no basis for Examiner's contention that "edges of an acyclic graph **inherently** are derived from a consistency constraint for publishing objects. Examiner's attempt to misconstrue

the teachings of Ferrel and the claimed inventions in an attempt to fit the claim language is improper as a matter of law, rendering the obviousness rejections legally deficient on their face.

E. **CONCLUSION**

Accordingly, for at least the above reasons, it is respectfully requested that the Board reverse all claim rejections under 35 U.S.C. §§ 102 and 103.



Frank V. DeRosa
Reg. No. 43,584

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889

APPENDIX A

1 ~ 15. (Canceled)

16. A method for publishing a plurality of objects comprising the steps of:

providing a plurality of objects, including compound objects;

partitioning at least some of the plurality of objects into a plurality of groups such that if two compound objects are constructed from at least one common changed fragment, then the compound objects are placed in a same group; and

publishing all objects belonging to a same group together.

17. The method as recited in claim 16, wherein the step of publishing includes the step of:

for at least two of the plurality of groups, publishing all objects belonging to a first group before publishing any objects belonging to a second group.

18. The method as recited in claim 16, wherein the step of publishing includes the step of:

delaying publication of a first object until a second object, which is referenced, by the first object is published.

19. (Original) The method as recited in claim 18, wherein the first and the second objects are Web pages and a reference between the first and the second objects is a hypertext link.

20. (Previously Presented) The method as recited in claim 16, further comprising the steps of:

representing at least some of the plurality of objects by nodes on at least one graph; and

representing one or more relationships between the objects by edges between the nodes.

21. (Previously Presented) The method as recited in claim 20, wherein the at least one graph includes an edge between two nodes representing compound objects if the two compound objects are constructed from at least one common changed fragment.

22. (Previously Presented) The method as recited in claim 20, wherein the edges include a directed edge from a first node representing a first object to a second node representing a second object, if the second object includes a reference to the first object.

23. (Previously Presented) The method of claim 20, further comprising the steps of:

determining if a first compound object and a second compound object embed at least one common changed fragment by:

topologically sorting at least part of the at least one graph;

examining the at least one graph in an order defined by the topological sort; and

when a node n1 is examined, for a node n2 which has changed and for which an edge from node n2 to node n1 exists, constructing a union between a set including node n2 and a set including changed fragments used to construct node n2.

24. (Original) The method as recited in claim 20 further comprising the step of performing a topological sort on at least part of the at least one graph for finding strongly connected components.

25. (Previously Presented) The method as recited in claim 24, further comprising the step of publishing a set of objects belonging to a same strongly connected component, of the at least one graph, together.

26. (Original) The method as recited in claim 24, further comprising the steps of:

examining objects in an order defined by the topological sort;

when an unpublished object is examined, publishing the unpublished object together with all objects belonging to a same strongly connected component.

27 ~ 41. (Canceled)

42. (Original) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for publishing a plurality of objects, the method steps comprising:

providing a plurality of objects, including compound objects;

partitioning at least some of the plurality of objects into a plurality of groups such that if two compound objects are constructed from at least one common changed fragment, then the compound objects are placed in a same group; and

publishing all objects belonging to a same group together.

43. (Original) The program storage device as recited in claim 42, wherein the step of publishing includes the step of:

for at least two of the plurality of groups, publishing all objects belonging to a first group before publishing any objects belonging to a second group.

44. (Original) The program storage device as recited in claim 42, wherein the step of publishing includes the step of:

delaying publication of a first object until a second object which is referenced by the first object is published.

45. (Original) The program storage device as recited in claim 44, wherein the first and the second objects are Web pages and a reference between the first and second objects is a hypertext link.

46. (Previously Presented) The program storage device as recited in claim 42, further comprising the steps of:

representing at least some of the plurality of objects by nodes on at least one graph; and

representing one or more relationships between the objects by edges between the nodes.

47. (Previously Presented) The program storage device as recited in claim 46, wherein the at least one graph includes an edge between two nodes representing compound objects if the two compound objects are constructed from at least one common changed fragment.

48. (Currently Amended) The program storage device as recited in claim 46, wherein the edges include a directed edge from a first node representing a first object to a second node representing a second object, if the second object includes a reference to the first object.

49. (Previously Presented) The program storage device of claim 46, further comprising the steps of:

determining if a first compound object and a second compound object embed at least one common changed fragment by:

topologically sorting the at least one graph;

examining the at least one graph in an order defined by the topological sort; and

when a node n1 is examined, for a node n2 which has changed and for which an edge from node n2 to node n1 exists, constructing a union between a set including node n2 and a set including changed fragments used to construct node n2.

50. (Original) The program storage device as recited in claim 46, further comprising the step of performing a topological sort on at least part of the at least one graph for finding strongly connected components.

51. (Previously Presented) The program storage device as recited in claim 50, further comprising the step of publishing a set of objects belonging to a same strongly connected component, of the at least one graph, together.

52. (Original) The method as recited in claim 50, further comprising the steps of:

examining objects in an order defined by the topological sort;

when an unpublished object is examined, publishing the unpublished object together with all objects belonging to a same strongly connected component.

53. (Previously Presented) A method for publishing a plurality of objects comprising the steps of:

providing a plurality of objects;

constructing at least one graph, the at least one graph including nodes representing at least some of the plurality of objects and edges for connecting nodes having relationships, at least some of the edges being derived from at least one consistency constraint;

finding at least one strongly connected component in the at least one graph; and

publishing a set of objects belonging to a same strongly connected component group.

54. (Canceled)

55. (Original) The method as recited in claim 53, further comprising the step of topologically sorting at least part of the at least one graph.

56. (Original) The method as recited in claim 55, further comprising the steps of:

examining objects in an order defined by topological sorting;

when an unpublished object is examined, publishing the unpublished object together with all objects belonging to a same strongly connected component.

57. (Previously Presented) The method as recited in claim 53, wherein the at least one consistency constraint includes delaying publication of a first object before a second object which is referenced by the first object is published.

58. (Original) The method as recited in claim 57, wherein the first and second objects include Web pages and at least one edge between the objects corresponds to at least one hypertext link.

59. (Original) The method as recited in claim 53, wherein an edge exists from a first object to a second object in at least one of the at least one graphs if the second object has a reference to the first object.

60. (Previously Presented) The method as recited in claim 53, wherein the at least one constancy constraint includes publishing two compound objects together if the two compound objects are both constructed from at least one common changed fragment.

61 ~ 74. (Canceled)

75. (Previously Presented) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for publishing a plurality of objects, the method steps comprising:

providing a plurality of objects;

constructing at least one graph, the at least one graph including nodes representing at least some of the plurality of objects and edges for connecting nodes having relationships, at least some of the edges being derived from at least one consistency constraint;

finding at least one strongly connected component in the at least one graph; and

publishing a set of objects belonging to a same strongly connected component group.

76. (Previously Presented) The program storage device of claim 75, wherein the methods steps further comprise topologically sorting at least part of the at least one graph.

77. (Previously Presented) The program storage device of claim 76, wherein the method steps further comprise:

examining objects in an order defined by topological sorting; and

when an unpublished object is examined, publishing the unpublished object together with all objects belonging to a same strongly connected component.

78. (Previously Presented) The program storage device of claim 75, wherein the method steps further comprise delaying publication of a first object before a second object which is referenced by the first object is published, based on the at least one consistency constraint.

79. (Previously Presented) The program storage device of claim 78, wherein the first and second objects include Web pages and at least one edge between the objects corresponds to at least one hypertext link.

80. (Previously Presented) The program storage device of claim 75, wherein an edge exists from a first object to a second object in at least one of the at least one graphs if the second object has a reference to the first object.

81. (Previously Presented) The program storage device of claim 75, wherein the method steps further comprise publishing two compound objects together if the two compound objects are both constructed from at least one common changed fragment, based on the at least one consistency constraint.

Evidence Appendix

There is no evidence submitted pursuant to 37 CFR §§ 1.130, 1.131 or 1.132 or any other evidence entered by the examiner and relied upon by appellant in this Appeal.

Related Proceedings Appendix

None.